2023

Selekcijski zadatak: Machine Learning inženjer

TIN POPOVIĆ

Analiza, obrada i priprema podataka

1. Što ste odlučili napraviti s null vrijednostima u skupu podataka? Koji su najčešći pristupi? Koji pristup je ovdje bolji i zašto?
   Na različite načine smo rukovali s nedostajućim vrijednostima u ovom koraku. Prvo, izbacili smo dva stupca koji su sadržavali preko 70% NaN vrijednosti („county“ i „size“). Zbog ogromne količine nevaljanih podataka u tim stupcima, nije imalo smisla tvrditi njihovu legitimnost te koristiti mehanizme zamijene NaN vrijednosti.
   Nadalje naš skup je sadržavao 5 varijabli ('year', 'model', 'fuel', 'odometer','transmission','title_status') koje su imali ispod 2% svojih zapisa ispunjenim NaN vrijednostima. Takve zapise smo jednostavno izbrisali , te smanjili broj naših zapisa s 426 880 na 405 594. Nakon toga, stanje našeg podatkovnog skupa u odnosu na broj nedostajućih vrijednosti je izgledao na sljedeći način:

```
Column: price  has:  0  NaN values.
That is 0.0 % of the DS

Column: year  has:  0  NaN values.
That is 0.0 % of the DS

Column: manufacturer  has:  15990  NaN values.
That is 3.942 % of the DS

Column: model  has:  0  NaN values.
That is 0.0 % of the DS

Column: condition  has:  162527  NaN values.
That is 40.071 % of the DS

Column: cylinders  has:  169060  NaN values.
That is 41.682 % of the DS

Column: fuel  has:  0  NaN values.
That is 0.0 % of the DS

Column: odometer  has:  0  NaN values.
That is 0.0 % of the DS

Column: title_status  has:  0  NaN values.
That is 0.0 % of the DS

Column: transmission  has:  0  NaN values.
That is 0.0 % of the DS

Column: drive  has:  121945  NaN values.
That is 30.066 % of the DS

Column: type  has:  88729  NaN values.
That is 21.876 % of the DS
```

   Varijabla „**manufacturer**“ sadrži oko 4% nedostajućih vrijednosti. Daljnjom analizom ove varijable utvrdili smo da ona ima preko 40 različitih vrijednosti od kojih je jedna „others“. Zbog neodređenosti tog zapisa, stupcima s NaN vrijednosti smo jednostavno dodijelili istu tu vrijednost „others“ te tako izgubili nedostajuće vrijednosti u ovoj varijabli. Također zbog jednostavnosti, dalje smo očuvali samo 15 najčešći proizvođača, a ostatku dodijelili vrijednost „others“.

   Varijabla „**cylinders**“ sadrži mnogo veći broj nedostajućih vrijednosti (preko 40%). Varijablu smo prvo pretvorili iz „object“ tipa u „int“ jer varijabla predstavlja broj cilindara. Zamjena

nedostajućih vrijednosti započela je prvo podjelom podatkovnog skupa po proizvođačima te pronalaskom najčešće vrijednosti broja cilindara za tog proizvođača. Tako je nedostajuća vrijednost zamijenjena s najčešćim brojem cilindra odgovarajućeg proizvođača.
Nedostajuće vrijednosti varijable „**drive**“ su zamijenjeni isto kao i varijabla „cylinders“. Prvo podjela po proizvođaču , te pronalazak najdominantnije vrijednosti varijable „drive“ te dodjela te vrijednosti NaN vrijednostima odgovarajućeg proizvođača.

Nedostajuće vrijednosti varijable „type“ su zamijenjeni s vrijednosti „other“, slično kao i varijabla „manufacturer“.

Za zamjenu nedostajućih vrijednosti varijable „**condition**“ koristi smo sljedeće korake. Podijeli smo podatkovni skupa na nove, srednje-dobi, i stare aute. Nadalje na temelju vrijednosti odometra smo podijelili ta tri podskupa. Ako je vrijednost odometra veća od medijana varijable te podgrupe onda je on klasificiran u „higherODM“, u protivnom u „lowerODM“. Tako u svih 6 podgrupa pronađen je mode te varijable, te nedostajuće vrijednosti su zamijenjene s njima u odnosu na starost i odometar automobila.

2. Ako ste odbacili neke kolone, navedite koje i objasnite zašto ste ih odbacili.
   Odbacio sam sljedeće stupce iz podatkovnog skupa:
   ['id', 'VIN','county','size','state','url','region',
   region_url','posting_date','image_url','paint_color','description','lat','long']
   Dio njih sam izbacio jer su sadržavali velik broj null vrijednosti, a ostatak jer smatram da ne mogu puno pridonijeti izradi prediktivnog modela. Identifikacijske varijable kao što su VIN, ID ne nose sa sobom informacije koje mogu doprinijeti predviđanju cijene automobila. Isto vrijedi i za ostale varijable koje su izbačene. S druge strane, imamo puno varijabli te potrebno napravit kompromis i zadržati par onih za kojih smatramo da su bitniji.

3. Ako ste izrađivali nove značajke, opišite ih i objasnite zašto ste ih dodali.
   Dodao sam novu značajku „age“ koja opisuje starost automobila. Dodao sam ju kako bi obogatio podatkovni skup. Postojala je velika korelacija između starosti i odometra, te sam stvorio novu varijablu „age*odometer“ i izbacio stupac „odometar“ kako bi izbjegao visoku korelaciju između značajki

4. Ako ste izrađivali vizualizacije, opišite zašto ste ih radili i kako one doprinose razumijevanju skupa podataka.
   Radio sam box-plot grafove za varijablu cijene i odometra. Radio sam ih da bi se riješili stršećih vrijednosti u našem skupu. Pored box-plota, napravio sam i korelacijsku matricu kako bi bolje ispitao ovisnosti između značajki.

5. Koje su najčešće podjele ukupnog skupa podataka u fazi pripreme za treniranje i evaluaciju modela? Kako ste vi pripremili skup podataka za sljedeći korak?
   Najčešća podjela podatkovnog skupa je na skup za treniranje i testiranje u omjeru 70:30. Podatkovni skup je prošao kroz niz koraka obrade i analize koji su razrađeni u Jupyter bilježnici. Izbrisane su nedostajuće i stršeće vrijednosti, pronađeni najbolji hiperparametri na skupu za

treniranje… (sve je opisano u Jupyter bilježnici). Pored podijele na 70:30 omjer moguće je isto uvrstiti validacijski set za pronalazak optimalnog modela.

Odabir, treniranje i evaluacija modela

1. Kako se formalno naziva kategorija problema koju rješavamo u ovom zadatku? Kako se razlikuje od klasifikacije?
   Formalni naziv ovog problema je regresija. Razlikuje se od klasifikacije jer predviđamo numeričku vrijednost.
2. Koje ste modele uzeli u obzir prilikom istraživanja? Koji ste model na kraju izabrali i zašto?
   Modeli koje sam koristio u ovoj fazi su: Linearna regresija, Lasso regresija, Decision Tree Regressor, Random Forest Regressor, KNN regressor i ElasticNet Regressor. Odlučio sam se za KNN Regressor jer je imao najbolji R2 rezultat te najmanji MSE.
3. Postoje li u vašem skupu podataka kategoričke varijable? Ako da, kako odabrani model funkcionira s njima? Jeste li morali kategoričke varijable dodatno obraditi kako bi ih model mogao iskoristiti?
   Da u mom skupu postoje kategoričke varijable, te ih je bilo potrebno detaljno obraditi kako bi ih model mogao koristiti. Obrađeni su putem One-Hot-Encoding i Ordinal-Encoding operacija.
4. Kako ste evaluirali performanse modela? Koje ste metrike koristili? Kakvi su rezultati vašeg modela?
   Model je evaluiran isključivo nad testnom skupu podataka, te rezultati evaluacije su zabilježeni kroz sljedeće metrike: MSE (Mean Squared Error), R2 score, RMSE i MAE.
   Rezultati modela su sljedeći:

	R2	MSE	MAE
Linear Regression	0.658	48947515.74	5253.89
Lasso	0.658	48947445.27	5252.91
Ridge	0.658	48947504.22	5253
Decision Tree R	0.806	27822464.75	3402.7
Random Forest R	0.838	23255159.57	3199.66
KNN R	**0.857**	**20424662.31**	**2361.02**
ElasticNet R	0.651	49927801.27	5314.18

5. Na koji ste način i u kojem formatu spremili model za buduće korištenje? Postoje li neki drugi načini?
   Model sam spremio putem joblib.dump() funkcije. Spremljen je u binarnom formatu s ekstenzijom .pkl. Postoje i drugi oblici spremanja modela, gdje možemo spremiti arhitekturu modela. Takav pristup je više karakterističan za neuronske mreže te nije primjenjiv na naš problem.

Izgradnja i kontejnerizacija API-a

1. U kojem formatu se šalju podatci o automobilu na vaš API? Što ako su poslani podatci neispravno formatirani? Što ako neka od vrijednosti nedostaje? Što ako su podatci poslani u krivom redoslijedu?
   Na API podaci se šalju u JSON obliku. Napravljeno je korisničko sučelje koje od korisnika traži input za sve varijable koje se koristi kao prediktori za model strojnog učenja kojeg smo trenirali u prethodnom koraku. Također kontroliran je i tip podataka kojeg korisnik želi poslati. U slučaju ako korisnik upiše nepoznatog proizvođača, ta vrijednost će se samo klasificirati u „others“. Tako ne trebamo brinuti o nedostajućim vrijednostima i neispravnom formatu podatka. Korisničko sučelje kontrolira red podatka koji se dalje šalje na API.

2. Koje su prednosti kontejnerizacije rješenja? Kako ste ugradili vaš spremljeni model u kontejner?
   Prednosti kontejnerizacije su sljedeće:
   - Lako reproduciranje: Docker omogućuje jednostavno reproduciranje okruženja, što znači da će aplikacija raditi na isti način bez obzira gdje se izvršava.
   - Izolacija: Kontejneri omogućuju izolaciju aplikacija i njihovih ovisnosti. Svaka aplikacija pokrenuta u Docker kontejneru ima svoje izolirano okruženje, što sprječava konflikte između različitih aplikacija i olakšava upravljanje ovisnostima.
   - Brzina: Kontejneri su brzi za pokretanje i zaustavljanje.

   Kontejnerizacija rješenja je ovdje implementirana putem Dockera. Prvo je napravljen Dockerfile koji se koristi za definiranje i konfiguraciju Docker kontejnera, te dodana je „requirements“ datoteka koja sadrži potrebne biblioteke za pokretanje ove aplikacije. Nakon toga izgrađen je Docker image , i rješenje je spremno za pokretanje u bilo kojem okruženju.

   Za pokretanje aplikacije potrebno je izvršiti sljedeću komandu:

   docker build -t flask-app .

   docker run -p 5000:5000 flask-app